別冊プログラムエリア

Short

PROGRAM
ショート プログラムライブラリ
LIBRARY

MSXマガジン11月号別冊付録

●STORY

地球人が自らを人と呼ぶように、ここガブ星には自らをシリガルと呼ぶ知的生命体が生息していた。知的とはいってもまだ本当に幼い文明であり、彼らは幾つかの種族に分かれ、その種族ごとに異なった体の色をしていた。そしてその色によって、イエロー族、レッド族などと呼ばれていた。そんなある日、イエロー族では古来から続く伝統儀式である成人式を行う日になった。この儀式は部屋にまかれたドットをすべて食べ尽くすという単純なものなのだがしかし、こともあろうにこの神聖なる儀式にレッド、ピンク、グリーン族のシリガルが邪魔に入ってきたのである。イエロー族はこれらの種族よりも弱い方に属する種族であったが、彼らから身を守る方法が一つだけあった。それはイエロー族のものがビックチップを食べることによって生じる独得の体臭であった。この体臭をかいだ彼らは急激に弱体化し、その状態で彼らに触れると、彼らをしばらくの間気絶させることができる。しかしこの効果はほんの数秒間しか持続しない。さあ、イエロー族の運命はいかに!

●遊び方

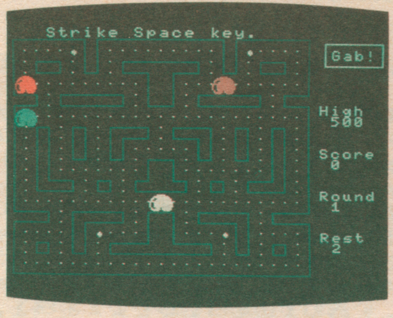
あなたはイエロー族(黄色)のシリガルをカーソルキーもしくはポート1に接続したジョイステックのいずれかを使って操作してください。もし二つ同時に操作したときはカーソルキーを優先的に扱います。そうして、画面内のすべてのドットとビックチップを食べ尽くすことによって、そのラウンドが終了します。通路にまかれている黄色いもののうち、小さいものがドットで、大きいものがビックチップです。また通常の状態で、レッド、ピンク、グリーン族のシリガルに触れると死亡となり残数が減ります。残数がなくなったりスコアが9998点をこえたりしたとき、GAME OVERとなります。しかし、イエロー族がビックチップを食べ敵が弱体化している間に彼らに触れると、この敵は気絶状態となりそこで一定時間動きを止めます。なお、彼らは弱体化しているあいだは水色に変わっています。

●敵の移動処理について

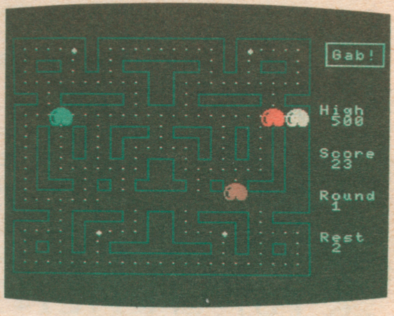
レッド族のシリガルは、まずイエロー族の横の位置に合わせるために、横の移動方向を算出します〔58行〕。ちなみに縦の移動方向

ヒップマン

は0です。そしてこの移動方向に進んだところ、つまり移動先が以下の条件を満たしているとき、そこに移動することができます。その条件の一つ目は、その移動方向が今までの移動方向に対して正反対の方向、つまりUターンになっていないことです。このプログラムにおいて唯一Uターンを行うのは、弱体化から通常に戻るときだけです。二つ目は、その移動先が壁以外のものであるということです。以上二つの条件が満たされている場合、そこに移動することができます〔59行〕。移動することができなかった場合、イエロー族の縦の位置に合わせるために、前と同様のことを縦と横とを逆にして考えます〔60～61行〕。これでもなお、進めなかった場合はランダムに移動します。このランダム処理は、はじめに四方のうち適当な方向を見ます。そして、これについても今までと同じように前記の二つの条件を満たしている場合に限り、そこに移動することができるようにします。これを時計回りに次々と方向を変え、移動先が見つかるまで同じことを繰り返します〔67～74行〕。このようにして、レッド族のシリガルは移動します。次にピンク族についてですが、彼はレッド族と異なり縦方向から合わすことを考えてから、横方向を合わすことを考え、最後にランダム処理をします。そして、グリーン族と弱体化した敵は、ランダム処理のみ行います。これが敵の移動処理の方法のすべてです。



☗赤も紫もとりあえず追いかけてくるぞ。



☗ただUターンはしないからそこを読もう。

●主な変数リスト

- H　ハイスコア
- S　スコア
- R　残数
- G　ラウンド数
- N　イエロー族の生存フラグ（生＝1、死＝0）
- Q　弱体化カウンター
  これが0より大なら敵は弱体化中
- L　敵停止カウンター
  この値の回数分のイエロー族の移動につき1回敵は移動をしない
- O　気絶させられる敵の点数Xイエロー族の横の位置
- Y　イエロー族の縦の位置
- M（x、y）部屋の構造
  （通路＝0、ビックチップ＝1、ドット＝2、壁＝3）
- A（i）敵の横の位置
- B（i）敵の縦の位置
- C（i）敵の横の移動方向
  （左＝－1、右＝1、現在地＝0）
- D（i）敵の横の移動方向
  （左＝－1、右＝1、現在地＝0）
  気絶状態なら気絶カウンターとして使用し、0より大の間は気絶状態とする。
- E（i）敵のシリガルの状態
  ＝0．通常のレッド族
  1．通常のピンク族
  2．通常のグリーン族
  3．弱体化中
  4．気絶状態
- F（i）敵の体の色
  i＝敵の状態
  i＝0のときレッド族
  1のときピンク族
  2のときグリーン族

●プログラムの構造

- 10　画面初期設定
- 11～13　キャラクターパターン設定
- 14　ぺーしっ君作動開始
- 15～16　変数初期設定
- 17～19　スプライトパターン設定
  敵の体の色とハイスコア初期値の設定
- 21　スタート時の変数初期設定
  （リプレイ時の飛び先）
- 22　画面表示（ラウンド更新時の飛び先）
- 23～27　敵の初期設定
- 28～32　スペースキー入力待機、ここでESCキーが押されたら、プログラム実行中止
- 33～37　キー入力処理、移動方向算出
- 38～42　移動先の様子による条件分岐
- 39　空所時の移動処理
- 40　ビックチップ時の敵の弱体化専用の移動処理
- 41～42　ドットとビッグチップ時の移動処理
- 43　レッド族の表示
- 44～49　接触処理
- 50　弱体化カウンターの1減
  ただし、0なら0のまま
  敵にやられたら85へ
- 51　ウェイト
- 52　敵停止カウンターが0なら32へ
- 53～77　敵の移動処理
- 54　敵がレッド族なら58へ
  ピンク族なら63へ
  グリーン族なら67へ
  気絶中なら56へ
- 55　まだ弱体化カウンターが0でないなら67、0なら57へ
- 56　気絶状態のままなら、気絶カウンターを1減し、77へ
- 57　通常の状態に戻す、76へ
- 58～62　レッド族専用の移動処理
- 63～66　ピンク族専用の移動処理
- 67～74　ランダム移動処理
- 75　移動先算出
- 76　敵の表示
- 78～83　接触処理
- 84　生存中なら32へ
- 85～88　死亡時の処理
- 89　9998点超えのメッセージ
- 90～91　ハイスコア処理
- 92～93　ゲームオーバー処理
- 94～97　ラウンドクリア、スコア処理
- 98～108　右部画面表示サブルーチン
- 109～112　部屋画面表示サブルーチン
- 113～135　画面データ
- 136～139　スプライトパターンデータ
- 140～147　キャラクターパターンデータ

ヒップマン

```
10 SCREEN1,2,0:COLOR11,1,1:KEY OFF:WIDTH32
11 RESTORE140:FORI=1 TO2:READ X,Y,Z
12 VPOKEBASE(6)+INT(X/8),Z:FORJ=8*X TO7+8*(X+Y)
13 READU:VPOKEBASE(7)+J,U:NEXTJ,I
14 CALL TURBO ON
15 DEFINTA-Z
16 DIM A(2),B(2),C(2),D(2),E(2),F(3),M(24,22)
17 RESTORE136
18 A$="":FORI=0 TO31:READJ:A$=A$+CHR$(J):NEXTI
19 FORI=0 TO3:SPRITE$(I)=A$:NEXTI
20 F(0)=7:F(1)=8:F(2)=13:F(3)=3:H=500
21 G=1:R=2:S=0:I=RND(-TIME)
22 PUT SPRITE0,(0,208):GOSUB98:GOSUB109
23 FORI=0 TO2:C(I)=0:D(I)=0:E(I)=I+1
24 J=INT(RND(1)*8):A(I)=1+J*3-(J>3)
25 J=INT(RND(1)*4):B(I)=1+J*3-(J=0)*20
26 PUT SPRITE1+I,(A(I)*8,B(I)*8),F(I+1):NEXTI
27 Q=0:L=1:N=1:X=12:Y=15:PUT SPRITE0,(96,120),10
28 LOCATE4,0:PRINT"Strike Space key.":LOCATE4,0
29 IFINKEY$=CHR$(27) THENEND
30 IFSTRIG(0)+STRIG(1)=0 GOTO29
31 PRINTSTRING$(17," ")
32 ONSTICK(0) GOTO34,35,35,35,36,37,37,37
33 ONSTICK(1)+1 GOTO44,34,35,35,35,36,37,37,37
34 V=0:W=-1:GOTO38
35 V=1:W=0:GOTO38
36 V=0:W=1:GOTO38
37 V=-1:W=0
38 ONM(X+V,Y+W) GOTO40,41,44
39 X=X+V:Y=Y+W:GOTO43
40 FORI=0 TO2:E(I)=-(E(I)=4)*4:NEXTI:Q=50:O=10
41 P=P-1:S=S+1
42 X=X+V:Y=Y+W:M(X,Y)=0:LOCATEX+1,Y+1:PRINT" ";
43 PUT SPRITE0,(X*8,Y*8),10
44 FORI=0 TO2:ONE(I) GOTO48,48,48,49
45 IFABS(X-A(I))+ABS(Y-B(I))>1 GOTO49
46 E(I)=4:D(I)=35:S=S+O:O=O+O
47 PUT SPRITE1+I,(A(I)*8,B(I)*8),4:GOTO49
48 IFABS(X-A(I))+ABS(Y-B(I))<2 THENN=0
49 NEXTI:GOSUB108
50 Q=Q-1-(Q=0):IFN=0 GOTO85 ELSE IFP=0 GOTO94
51 FORI=0 TO1499-(G<11)*(11-G)*90:NEXTI
52 L=L-1:IFL=0 THENL=INT(G/2)+3:GOTO32
53 FORI=0 TO2
54 U=A(I):Z=B(I):ONE(I) GOTO58,63,67,56
55 IFQ GOTO67 ELSE57
56 IFD(I) THEND(I)=D(I)-1:GOTO77
57 C(I)=0:D(I)=0:E(I)=I+1:U=A(I):Z=B(I):GOTO76
58 V=SGN(X-U):W=0
59 IFV THEN IFC(I)<>-V ANDM(U+V,Z)<3 GOTO75
60 V=0:W=SGN(Y-Z)
61 IFW THEN IFD(I)<>-W ANDM(U,Z+W)<3 GOTO75
62 GOTO67
63 V=0:W=SGN(Y-Z)
64 IFW THEN IFD(I)<>-W ANDM(U,Z+W)<3 GOTO75
65 V=SGN(X-U):W=0
66 IFV THEN IFC(I)<>-V ANDM(U+V,Z)<3 GOTO75
67 J=INT(RND(1)*4)+1
68 ONJ GOTO69,70,71,72
69 V=0:W=-1:J=2:GOTO73
70 V=1:W=0:J=3:GOTO73
71 V=0:W=1:J=4:GOTO73
72 V=-1:W=0:J=1
73 IFM(U+V,Z+W)=3 GOTO68
74 IFC(I)=-V ANDD(I)=-W GOTO68
75 A(I)=U+V:B(I)=Z+W:C(I)=V:D(I)=W
76 PUT SPRITE1+I,(A(I)*8,B(I)*8),F(E(I))
77 NEXTI
78 FORI=0 TO2:ONE(I) GOTO82,82,82,83
```

```
79 IFABS(X-A(I))+ABS(Y-B(I))>1 GOTO83
80 E(I)=4:D(I)=35:S=S+O:O=O+O
81 PUT SPRITE1+I,(A(I)*8,B(I)*8),4:GOTO83
82 IFABS(X-A(I))+ABS(Y-B(I))<2 THENN=0
83 NEXTI
84 IFN GOTO32
85 FORI=15 TO4 STEP-1:PUT SPRITE0,(X*8,Y*8),I
86 FORJ=0 TO1599:NEXTJ,I
87 R=R-1:IFR<0 GOTO90
88 GOSUB106:GOTO23
89 S=9999:LOCATE4,14:PRINT"How great you are!"
90 IFS<=H GOTO92
91 H=S:LOCATE4,11:PRINT"Your score is best."
92 GOSUB107:LOCATE9,9:PRINT"GAME OVER"
93 FORI=0 TO69:FORJ=0 TO1999:NEXTJ,I:GOTO21
94 LOCATE5,9:PRINTG;"Round Clear!"
95 S=S+R*10+G*5:IFS>9998 GOTO89
96 GOSUB108:FORI=0 TO99:FORJ=0 TO1999:NEXTJ,I
97 G=G+1:GOTO22
98 CLS:LOCATE26,1:PRINT"┌────┐"
99 LOCATE26,2:PRINT"|Gab!|"
100 LOCATE26,3:PRINT"└────┘"
101 LOCATE26,15:PRINT"Round"
102 LOCATE26,19:PRINT"Rest"
103 LOCATE26,7:PRINT"High"
104 LOCATE26,11:PRINT"Score"
105 LOCATE26,16:PRINT G
106 LOCATE26,20:PRINT R
107 LOCATE26,8:PRINT H
108 LOCATE26,12:PRINT S:RETURN
109 RESTORE113:FORJ=0 TO22:READA$:LOCATE1,J+1
110 PRINTA$;:FORI=0 TO24:B$=MID$(A$,I+1,1)
111 M(I,J)=3+(B$="タ")+(B$="チ")*2+(B$=" ")*3
112 NEXTI,J:P=224:RETURN
113 DATA"アイイイイイウアイイイイイイイイイウアイイイイイウ"
114 DATA"ウタタタタチウウタタタタタタタタタウウチタタタタウ"
115 DATA"ウタアイウタウウタアイイイイイウタウウタアイウタウ"
116 DATA"ウタイイエタイエタイイウ アイエタイエタイイエタウ"
117 DATA"ウタタタタタタタタタタタウ ウタタタタタタタタタタタウ"
118 DATA"イイウタアイイイイウタウ ウタアイイイイウタアイエ"
119 DATA"アイエタイイイイイエタイイエタイイイイイエタイイウ"
120 DATA"ウタタタタタタタタタタタタタタタタタタタタタタタウ"
121 DATA"ウタアウタアウタアイイウタアイイウタアウタアウタウ"
122 DATA"ウタウウタウウタウアイエタイイウウタウウタウウタウ"
123 DATA"ウタウウタウウタウウタタタタタウウタウウタウウタウ"
124 DATA"ウタウウタイエタイエタアイウタイエタイエタウウタウ"
125 DATA"ウタウウタタタタタタタウ ウタタタタタタタウウタウ"
126 DATA"ウタウイイウタアウタアエ イウタアウタアイエウタウ"
127 DATA"ウタイイイエタイエタイイイイイエタイエタイイイエタウ"
128 DATA"ウタタタタタタタタタタタ タタタタタタタタタタタウ"
129 DATA"イイイウタアイイイウタアイウタアイイイウタアイイエ"
130 DATA"アイイエタウアイイエタウ ウタイイイウウタイイイウ"
131 DATA"ウタタタタウウチタタタウ ウタタタチウウタタタタウ"
132 DATA"ウタアウタウウタアイイエ イイイウタウウタアウタウ"
133 DATA"ウタイエタイエタイイイイイイイイエタイエタイエタウ"
134 DATA"ウタタタタタタタタタタタタタタタタタタタタタタタウ"
135 DATA"イイイイイイイイイイイイイイイイイイイイイイイイエ"
136 DATA0,0,3,15,31,51,39,127
137 DATA79,79,79,79,39,63,30,0
138 DATA0,0,224,248,252,254,254,255
139 DATA255,255,255,255,126,126,60,0
140 DATA177,3,65
141 DATA255,128,128,128,128,128,128,128
142 DATA255,0,0,0,0,0,0,0
143 DATA128,128,128,128,128,128,128,128
144 DATA128,0,0,0,0,0,0,0
145 DATA192,1,161
146 DATA128,0,0,0,0,0,0,0
147 DATA64,224,224,64,0,0,0,0
```

ブロックくずし

●遊び方

もう説明の必要もないでしょうが、TVゲームの古典中の古典、ブロックくずしです。このプログラムはすべてBASICで書かれていますが、けっこう速くうごきますよ。一応ゲームの説明をしますと、ボールをうちかえして画面上のすべてのブロックを破壊してください。すると次の面に進みますので、しつこくブロックを破壊して下さい。あなたはそれをゲームオーバー、つまりボールをうけそこなうまで続けてください。さて、操作法ですが、自分はカーソルキーで左右に動きます。また、スペースキーを押しながらカーソルキーを押すと自分の動きのスピードが倍になります。いざというとき使ってください。ふだんからスペースキーを押しっぱなしだと動きがはやすぎてかえってやりずらいと思いますよ。

さて、ゲームの説明はここまでです。遊ぶだけならここまで読めばOKです。けれど、「このプログラムはどうやって動いているの?」とか、「このプログラムをいじってみたい」とか、「自分でゲームを作る時の参考にしたい」とか思ってくれる人もいるかもしれませんので、これからプログラムの説明をしましょう(なんて親切なんだ!)。

●プログラムの説明

このプログラムのポイントは、ボールがいったい何にぶっかったのかを見分けることにあります。見たところ、かべも、ブロックも、自分もみんな同じ正方形のキャラクタでしょう。ところがほんとは違うんです。もとは、かべが〝*〟、自分は〝=〟、ブロックは〝+〟、下のかべ(つまりここにボールが当たるとゲームオーバー)は〝/〟というキャラクタだったのです。さて、MSXのSCREEN 1では、キャラクタを自由に書きかえることができるのを知っていますか?
ためしにSCREEN 1でVPOKE&H100、&HFF↵とやってみてください。どうなりましたか? べつにあなたのMSXが故障したわけではありませんから心配しないでください。これはCHR$(&H20)、つまりスペースのキャラクタを書きかえたためごらんのようになったのです。このゲームでもこれと同じように〝*〟、〝=〟、〝+〟、〝/〟をみんな正方形のキャラクタに書きかえてしまったわけです。こうすれば、ボールが一見同じに見える何にあたったのかを判断することができるわけです。。変な小細工をしているのはこれくらいで、他に気を使った点といえばプログラムのループ(何回もくりかえす部分)をできるだけ短くして、スピードを落とさないよう注意したというくらいです。実際、メインとなる部分は7行しかありません。また、BASICでゲームを作る場合、スピードをできるだけ速くすることは重要ですが、そのためには処理に時間のかかるIF文をあまり通らないようにするのはポイントのひとつです。このプログラムのループ中では、「ボールが何かにあたったか?」「カーソルキーが押されたか?」というふたつのIF文しかありません。

●プログラムの構造

それでは、プログラムの細かい部分の説明をしましょう。

100〜110
初期設定です。

120〜170
*、+、/、=のキャラクターを正方形のブロックに「化け」させるルーチンです。

180
F1キーの割り込みの設定です。ここまで読んでくれた人だけにおしえちゃいますけど、実はF1キーを押すと次の面にすすむことができるんですよ!

190
OFC=VRAMの先頭アドレス。

200〜220
PSGの初期設定です。

230〜250
SC$(N)に1〜8面のパターンを読みこみます。9面以降は同じパターンのくりかえしです。

260〜290
スプライト(ボール)の設定です。

300
ハイスコアの初期設定です。

310〜340
「TRY AGAIN?」でYを押したときここに飛んできます。SC=0(面=1)、S1=0(スコア=0)と設定します。

350〜470
ブロックやかべなど、初期画面を描きます。

480〜490
自分(KX)やボール(X、Y)の座標の設定、ボールの移動方向(DX、DY)、ブロックを破壊した数(H1)のクリアをします。

500〜560
自分やボール、スコアなどの初期値を表示します。

570〜640
メインループ。前に説明したとおりです。カーソルキーが押されていれば1080へ、ボールが何かにぶつかれば660へ飛びます。

650〜700
ボールが何にあたったかを判断します。

710〜760
ボールがかべにあたったときの反射の処理をします。

770〜890
ブロックにあたったとき、ブロックを消したり、ボールを反射させたり、スコアを計算したりの処理をします。

900〜1010
ボールをうけそこねたとき、つまりゲームオーバーの処理です。

1020〜1060
ボールをうけたときの反射の処理をします。

1070〜1120
カーソルキーが押されたとき、自分を動かします。

1230
正方形に「化け」させるキャラクタの一覧です。

1240
PSGに書きこむデータです。

1250〜1320
一目瞭然、パターンのデータです。当然、ここを書きかえればパターンをかえることができます。どう書きかえるかって? そんなの見りゃわかるでしょ!

1330
ボールのキャラクタのデータです。

1350〜1390
これが幻の隠しコマンド!必殺パターンスキップだ!

というわけで、おおまかですが、プログラムの説明をおわります。あとはあなたが実際にプログラムを追ってみてください。

```
100 DIM SC$(7)
110 SCREEN 1:COLOR 15,12,12:WIDTH 32:KEY OFF
120 FOR N=1 TO 4
130  READ A$:AD=ASC(A$)*8
140  FOR M=AD TO AD+7
150   VPOKE M,&HFF
160  NEXT M
170 NEXT N
180 ON KEY GOSUB 1350:KEY(1) ON
190 OFS=&H1800
200 FOR R=0 TO 13
210  READ X:SOUND R,X
220 NEXT R
230 FOR N=0 TO 7
240  READ SC$(N)
250 NEXT N
260 FOR N=1 TO 8
270  READ A$:SP$=SP$+CHR$(VAL("&H"+A$))
280 NEXT N
290 SPRITE$(0)=SP$
300 HI=1000
310 '
320 SC=0:S1=0
330 SOUND 7,&H3E:SOUND 12,40
340 CLS:VPOKE &H2005,&H7C:VPOKE &H2010,&H9C
350 'Screen Define
360 CN=0
370 FOR Y=1 TO 9
380  IF MID$(SC$(SC MOD 8),Y,1)="*" THEN CN=CN+6
:LOCATE 5,Y:PRINT"♣♣♣ ♣♣♣ ♣♣♣ ♣♣♣ ♣♣♣ ♣♣♣";
390 NEXT Y
400 FOR X=2 TO 30
410  LOCATE X,0:PRINT"*";
420  LOCATE X,22:PRINT"/";
430 NEXT X
440 FOR Y=1 TO 21
450  LOCATE 2,Y:PRINT"*";
460  LOCATE 30,Y:PRINT"*";
470 NEXT Y
480 X=12:Y=19:KX=10:H1=0
490 DX=1:DY=-1
500 LOCATE 3,23:PRINT"SCORE   00";
510 LOCATE 18,23:PRINT USING"HIGH #####";HI;
520 LOCATE KX,20:PRINT"===="
530 LOCATE 12,10:PRINT"SCENE ";SC+1
540  PUT SPRITE 0,(X*8,Y*8),15,0
550 FOR T=1 TO 500:NEXT T
560 LOCATE 12,10:PRINT"          "
570 'Main Loop
580 ST=STICK(0)
590  IF ST<>0 THEN GOSUB 1080
600  A=VPEEK((Y+DY)*32+X+DX+OFS)
610  IF A<>ASC(" ") THEN GOSUB 660
620  X=X+DX:Y=Y+DY
630  PUT SPRITE 0,(X*8,Y*8),15,0
640 GOTO 580
650 'Hit Something
660  IF A=ASC("*") THEN 720
670  IF A=ASC("♣") THEN 780
680  IF A=ASC("/") THEN 910
690  IF A=ASC("=") THEN 1030
700 RETURN
710 'Hit Wall
720 AX=VPEEK(Y*32+X+DX+OFS):AY=VPEEK((Y+DY)*32+X
+OFS)
730 IF AX=ASC("*") THEN DX=-DX
740 IF AY=ASC("*") THEN DY=-DY
750 SOUND 0,80:SOUND 13,0
```

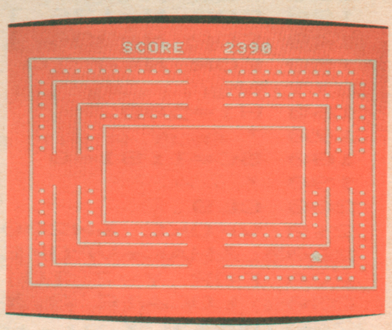
```
760 RETURN
770 'Hit Block
780 AX=VPEEK(Y*32+X+DX+OFS):AY=VPEEK((Y+DY)*32+X
+OFS)
790 H1=H1+1
800 LOCATE ((DX+X-5)¥4)*4+5,Y+DY:PRINT"    ";
810 DY=-DY
820 IF AY<>ASC("♣") THEN DX=-DX
830 SOUND 0,150:SOUND 13,0
840 S1=S1+100
850 LOCATE 3,23:PRINT USING"SCORE #####";S1;
860 IF S1>HI THEN HI=S1
870 LOCATE 18,23:PRINT USING"HIGH #####";HI;
880 IF H1=CN THEN LOCATE KX,20:PRINT"    ";:SC=S
C+1:RETURN 360
890 RETURN
900 'Miss boal
910 SOUND 7,&H37:SOUND 12,90:SOUND 13,0
920 FOR N=1 TO 10
930  COLOR ,,8:GOSUB 1010
940  COLOR ,,12:GOSUB 1010
950 NEXT N
960 LOCATE 10,10:PRINT"< GAME OVER >"
970 LOCATE 8,12:PRINT"TRY AGAIN?(y or n)"
980 A$=INKEY$
990 IF A$="y" THEN 320
1000 IF A$="n" THEN SCREEN 0:END ELSE 980
1010 FOR T=1 TO 20:NEXT T:RETURN
1020 'Hit Reflecter
1030 AX=VPEEK(Y*32+X+DX+OFS):AY=VPEEK((Y+DY)*32+
X+OFS)
1040 DY=-DY
1050 IF AY<>ASC("=") THEN DX=-DX
1060 RETURN
1070 'Move Refrecter
1080 IF ST=7 THEN 1120
1090 IF ST=3 THEN 1180
1100 RETURN
1110 'Move Left
1120 LOCATE KX,20:PRINT"    ";
1130 IF STRIG(0)=0 THEN KX=KX-1 ELSE KX=KX-2
1140 IF KX<3 THEN KX=3
1150 LOCATE KX,20:PRINT"====";
1160 RETURN
1170 'Move Right
1180 LOCATE KX,20:PRINT"    ";
1190 IF STRIG(0)=0 THEN KX=KX+1 ELSE KX=KX+2
1200 IF KX>26 THEN KX=26
1210 LOCATE KX,20:PRINT"====";
1220 RETURN
1230 DATA *,♣,=,/
1240 DATA 0,0,0,0,0,0,30,&h3e,16,0,0,0,40,0
1250 DATA --*-*----
1260 DATA -*-*-*---
1270 DATA -***-*---
1280 DATA -*-*-*-*-
1290 DATA *-------*
1300 DATA *---*---*
1310 DATA *****----
1320 DATA *-*-*-*-*
1330 DATA 3e,7f,7f,7f,7f,7f,3e,0
1340 '
1350 SC=SC+1
1360 FOR N=1 TO 10
1370  LOCATE 3,N:PRINT STRING$(25," ");
1380 NEXT N
1390 RETURN 360
```

ヘッドオン

●遊び方

さて、レトロブームにのってお送りするなつかしのゲーム第二弾は、ヘッドオンです。これもブロックくずし同様説明の必要がないくらい有名なゲームです。一応説明すると、敵（O）をよけながら点を消していってください。全部点をクリアすれば１面クリアです。敵の動きを読みながら、衝突しないようにうまくコースを選んでください。

それでは、プログラムの説明をします。このプログラムでも、ブロック崩し同様、キャラクタを化けさせるというテクニックを使っています。敵にしても自分にしてもコースに沿って進まなくてはいけませんが、そのためにはかどで曲がることが必要です。それではかどかどうかを見分けるにはどのようにしているのでしょうか。このプログラムではまわりのかべを見ながら進む、ということはしていません。では、どうやっているのかというと、曲がるべきところにはそれどれ「こっちへ曲がれ」というサインとなるキャラクタがかくされています。敵、あるいは自分がそのキャラクタを踏むと、そのキャラクタから曲がる方向を判断して、そちらの方向に進むのです。それでは、もう少し詳しく説明しましょう。



⬆何も置いていないように見える場所に隠されているキャラクタが高速化のカギだ。

●プログラムの構造

100
スクリーンモードの設定です。

110〜120
前で説明した、「曲がるサイン」の読み込みです。敵と自分では進む方向が逆なので、サインの解釈が違います。（XD（N)、YD（N））は敵、（XR（N)、YR（N））は自分の進む方角で、N＝0〜3までがそれぞれCHR＄（17）〜CHR＄（20）に対応しています。

130〜140
ブロック崩し同様、＊、┼、├、┴、┤、┬のキャラクタをブランクに書き換えます。

150
PSGの設定

160
ハイスコア（H1）の設定。

180
面数（SC)、スコア（S1）のクリアです。

200〜620
初期画面を描きます。

640〜660
変数の初期値の設定です。（X、Y）は敵の座標、（XM、YM）は自分の座標です。

670〜700
READY！

720
敵とぶつかったならゲームオーバーのルーチンへとびます。

730
敵がこれから進む場所のキャラクタをQへ読み込みます。

740
敵を表示。

750
かどだったら曲がります。

760〜830
敵のコース変更です。敵はコースが切れたところでは必ず動きます。たとえまっすぐ進むときも右に半歩、左に半歩、結果はまっすぐといったぐあいになりますので、まどわされないよう注意してください。

850
敵一歩に対して、自分は二歩進みます。

860
敵がこれから進む場所のキャラクタをQへ読み込みます。

870
自分を表示します。

880
かどだったら曲がります。

890
もし点だったら、点をクリアしてスコアを加算します。また、スコアが１面の点の数の倍数だったら１面クリアとみなし、1150行へとびます。

900〜1040
自分のコース変更です。もちろん、コースのない方向には進めません。また、半歩しか移動せず、先が道でない場合はそのままだとかべにささってしまうので、強制的にもう半歩移動させます。

1050
自分が移動してくる前にあったキャラクターを再表示します。

1060
自分が移動したとき、その方向を次に移動するときまで覚えておきます。これを使って、前に説明した強制移動をするのです。

1070
自分の座標の移動。

1080
敵とぶつかったとき、FORループの外でゲームオーバーの処理をするため、ループの変数Nの値を大きくしてループからぬけださせます。

1090
850行のFORに対するNEXT。

1100
もし、敵とぶつかっていたならゲームオーバーのルーチンへ。

1110
敵が移動してくる前にあったキャラクタを再表示します。

1120
敵の座標の移動。

1130
メインループは終わりです。

1150〜1170
１面クリアしたときの処理です。

1190〜1200
「サイン」に対する移動方向のデータです。

1210
PSGの初期データです。

1220
ただのタイマーです。

1240〜1360
ゲームオーバーの処理です。お疲れ様でした！

```
100 SCREEN 1:COLOR 15,12,12:WIDTH 32:KEY OFF
110 FOR N=0 TO 3:READ XD(N),YD(N):NEXT N
120 FOR N=0 TO 3:READ XR(N),YR(N):NEXT N
130 FOR N=136 TO 168:VPOKE N,0:NEXT N
140 FOR N=336 TO 352:VPOKE N,0:NEXT N
150 FOR N=0 TO 13:READ X$:SOUND N,VAL("&h"+X$):N
EXT N
160 H1=500
170 '
180 SC=1:S1=0
190 '
200 CLS
210 VPOKE &H2002,&H3C:VPOKE &H2003,&H3C:VPOKE &H
2005,&HAC:VPOKE &H2010,&H7C
220 FOR Y=2 TO 21
230  LOCATE 3,Y:PRINT STRING$(28,"・");
240 NEXT Y
250 FOR N=0 TO 6 STEP 2
260  FOR X=2+N TO 30-N
270   LOCATE X,1+N:PRINT"─";
280   LOCATE X,22-N:PRINT"─";
290  NEXT X
300 NEXT N
310 FOR N=0 TO 6 STEP 2
320  FOR Y=1+N TO 21-N
330   LOCATE 2+N,Y:PRINT"|";
340   LOCATE 30-N,Y:PRINT"|";
350  NEXT Y
360 NEXT N
370 Y=1
380 FOR X=2 TO 8 STEP 2
390  LOCATE X,Y:PRINT"┌";
400  LOCATE X+1,Y+1:PRINT"├";
410  LOCATE 32-X,Y:PRINT"┐";
420  LOCATE 32-X-1,Y+1:PRINT"┬";
430  LOCATE X,23-Y:PRINT"└";
440  LOCATE X+1,23-Y-1:PRINT"┤";
450  LOCATE 32-X,23-Y:PRINT"┘";
460  LOCATE 32-X-1,23-Y-1:PRINT"┴";
470  Y=Y+2
480 NEXT X
490 FOR Y=2 TO 6
500  LOCATE 15,Y:PRINT"* *";
510  LOCATE 15,Y+15:PRINT"* *";
520 NEXT Y
530 LOCATE 3,10:PRINT"+++++";
540 LOCATE 3,11:PRINT"     ";
550 LOCATE 3,12:PRINT"+++++";
560 LOCATE 25,10:PRINT"+++++";
570 LOCATE 25,11:PRINT"     ";
580 LOCATE 25,12:PRINT"+++++";
590 FOR Y=8 TO 15
600  LOCATE 9,Y:PRINT STRING$(15," ");
610 NEXT Y
620 SOUND 7,&H3E:SOUND 12,&H10
630 '
640 OFS=&H1800
650 X=3:Y=2:XD=1:YD=0:XA=0:YA=0:XS=0:YS=0
660 XM=3:YM=21:XP=0:YP=0:XR=1:YR=0
670 FOR N=1 TO 5
680  LOCATE 14,10:PRINT"READY!";:GOSUB 1220
690  LOCATE 14,10:PRINT"      ";:GOSUB 1220
700 NEXT N
710 '
720 IF X=XM AND Y=YM THEN 1240
730 Q=VPEEK(Y*32+X+OFS)
740 LOCATE X,Y:PRINT"♣";
750 IF Q=>17 AND Q<=20 THEN XD=XD(Q-17):YD=YD(Q-
17)
760 IF Q<>42 THEN YS=0:GOTO 800
770 IF VPEEK((Y+1)*32+X+OFS)<>42 THEN YS=-1:GOTO
```

```
 800
780 IF VPEEK((Y-1)*32+X+OFS)<>42 THEN YS=1:GOTO
800
790 YS=SGN(RND(-TIME)-.5):IF YS=0 THEN 790
800 IF Q<>43 THEN XS=0:GOTO 850
810 IF VPEEK(Y*32+X+1+OFS)<>43 THEN XS=-1:GOTO 8
50
820 IF VPEEK(Y*32+X-1+OFS)<>43 THEN XS=1:GOTO 85
0
830 XS=SGN(RND(-TIME)-.5):IF XS=0 THEN 830
840 '
850 FOR N=1 TO 2
860 P=VPEEK(YM*32+XM+OFS)
870 LOCATE XM,YM:PRINT"*";
880 IF P=>17 AND P<=20 THEN XR=XR(P-17):YR=YR(P-
17):GOTO 1020
890 IF P=165 THEN SOUND 13,0:P=32:S1=S1+1:LOCATE
 10,0:PRINT USING"SCORE #####0";S1:IF S1 MOD 174
=0 THEN 1150
900 IF P<>42 THEN YP=0:GOTO 960
910 KY=STICK(0)
920 IF KY=1 THEN YP=-1
930 IF KY=5 THEN YP=1
940 IF VPEEK((YM+1)*32+XM+OFS)<>42 AND YP=1 THEN
 YP=0:GOTO 960
950 IF VPEEK((YM-1)*32+XM+OFS)<>42 AND YP=-1 THE
N YP=0
960 IF P<>43 THEN XP=0:GOTO 1020
970 KY=STICK(0)
980 IF KY=3 THEN XP=1
990 IF KY=7 THEN XP=-1
1000 IF VPEEK(YM*32+XM+1+OFS)<>43 AND XP=1 THEN
XP=0:GOTO 1020
1010 IF VPEEK(YM*32+XM-1+OFS)<>43 AND XP=-1 THEN
 XP=0
1020 P1=VPEEK((YM+YP+YR)*32+XM+XP+XR+OFS)
1030 IF P1=22 THEN XP=XP+XA:XA=0
1040 IF P1=23 THEN YP=YP+YA:YA=0
1050 VPOKE(YM*32+XM+OFS),P
1060 IF XP<>0 OR YP<>0 THEN XA=XP:YA=YP
1070 XM=XM+XR+XP:YM=YM+YR+YP
1080 IF X=XM AND Y=YM THEN N=9
1090 NEXT N
1100 IF N>3 THEN 1240
1110 VPOKE(Y*32+X+OFS),Q
1120 X=X+XD+XS:Y=Y+YD+YS
1130 GOTO 720
1140 '
1150 LOCATE 12,10:PRINT"SCENE";SC:SC=SC+1
1160 LOCATE 12,12:PRINT"COMPLATE!"
1170 FOR N=1 TO 6:GOSUB 1220:NEXT N
1180 GOTO 200
1190 DATA -1,0,0,1,0,-1,1,0
1200 DATA 0,-1,-1,0,1,0,0,1
1210 DATA 60,0,0,0,0,0,10,3e,10,0,0,0,10,0
1220 FOR TT=1 TO 100:NEXT TT:RETURN
1230 '
1240 SOUND 7,&H37:SOUND 12,120:SOUND 13,0
1250 FOR N=1 TO 5
1260  COLOR ,8,8:GOSUB 1360
1270  COLOR ,12,12:GOSUB 1360
1280 NEXT N
1290 LOCATE 12,10:PRINT"GAME OVER";
1300 IF S1>H1 THEN H1=S1
1310 LOCATE 4,21:PRINT USING"TODAY'S HIGH SCORE
:#####0";H1
1320 LOCATE 9,12:PRINT"TRY AGAIN?(y/n)";
1330 KY$=INPUT$(1)
1340 IF KY$="y" THEN 180
1350 IF KY$="n" THEN END ELSE 1330
1360 FOR TT=1 TO 10:NEXT TT:RETURN
```

BOMB!!

●STORY

彼は対潜哨戒機Q４Dのパイロットであった。ある日、彼の管轄の海域に正体不明の物体が侵入、彼はいつものようにQ４Dで出動した。しかしどうも相手の動きがおかしい。いったい何者なんだ？　さて、Q４Dのレーダーは旧式のオンボロで、しかも故障中である。なんと、敵の方向はわかっても敵までの距離がわからないのだ。おまけにあまり精度もよくないときている。「まったく、こんなもんのどこがレーダーなんだ！」と彼は舌を打った。しかし、そんな旧式のレーダーでも、海中をうごめく物体が潜水艦なんかじゃないということはすぐにわかった。追尾すればするほど彼はわけがわからなくなった。それではいったい………、無線機からの雑音混じりの叫び声で彼は我にかえった。「基地が全滅!?」Q４Dはあくまでもパトロール機であり、武器はほとんどもっていないが、なぜか爆弾を１個しょっている。普段はいたって平和なこの海域ではこんなものついてるだけムダだと思っていたのだが………、また、Q４Dはフロートを持っているので、着水することが可能である。「敵は１機か………やってみるか」彼は自分に問いかけるようにつぶやいた。つまり、１発の爆弾で敵をしとめ、着水しようというのだ。そのあとどうなるかはわからない。新たな敵がやってくるかもしれない。しかし燃料計は『EMPTY』に限りなく近づいている。もうあまりもたないだろう。彼に選択の余地はなかった。

●遊び方

というわけで、ゲームは始まります。あなたに与えられたものは１発の爆弾とオンボロのレーダーだけです。敵はただ一機ですが、１発必中あるのみ。もし爆弾が命中しなければそこでゲームオーバーです。さらに制限時間内にこの任務を完了させなければなりません。残り時間は画面の左上に表示されます。時間をオーバーすれば、やはりそこでゲームオーバーです。

このゲームは全部で10面あります。最初の面ではあなたに約80秒の時間が与えられます。１面進むごとに持ち時間は８秒ずつ減っていきます。つまり最後の面では８秒しかないことになります。さて、１面めは敵の座標に点が表示されます。いってみれば「練習モード」ですね、これでレーダーの向きと敵の位置関係をつかんでおいてください。この面は時間ギリギリまで使って動きを研究することをおすすめします。実際にやってみればレーダーが「オンボロ」だといった理由がわかると思いますよ。２面目から本番スタートです。もう点は表示されませんので、あなたの研ぎ澄まされた（？）カンで敵を撃破してください。

●プログラムの説明

それではプログラムの説明をしましょう。このゲームで登場するレーダーは、16パターンのスプライトで表示されています。グラフィックで線や円を書いて動かしていたのではおそくなるからです。もっとも、16＊16のスプライトは16種類しかパターンが作れないのでレーダーの精度が悪くなってしまったのですが、さて16＊16ドットのスプライトを16個作るとなると、512バイトのデータが必要ですが、このプログラムリストを見回してみてもそんなデータは見あたりません。それでは、どこからデータをとってくるかというと、このプログラムをRUNさせると画面の上の方に16パターンのレーダーが表示されます。実はVRAM上のこのパターンをスプライトに「コピー」しているのです。このように似たようなパターンがたくさん必要で、しかも簡単な命令で画面上に描ける場合、このようなテクニックを使えばデータを作るために方眼紙に点を打ったり、プログラム中にたくさんのデータを書き並べなくてもすむわけです。

他にはとくにかわったことはやっていませんが、ａｔｎθという関数は高校生以下の人にはなじみがないかもしれませんね。ａｔｎはｔａｎの逆関数と呼ばれるものなのです。例えば、ｔａｎ（π／４）＝１ということは三角関数を習った人なら知ってるはずですが、これに対してａｔｎ（１）＝π／４というわけです。もし、あなたがπの値をちょっとくわしく知りたいと思ったなら、PRINT ATN（１）＊４とキーをたたいてみてください。たぶんあなたが知っているよりも多くのπのけた数が表示されるでしょう。もっとも、あなたが何十けたもπを知っているのなら別ですけどね、それではもう少し詳しく説明しましょう。

●プログラムの構造

100〜110
スクリーンモードの設定です。

120
ＳＴＩＣＫ（０）の値に対してのＸ、Ｙの移動値をＸＭ（Ｎ）、ＹＭ（Ｎ）に読み込みます。

130
ＰＳＧの初期設定です。

140〜260
レーダーのスプライトの定義です。

270
ハイスコアの初期値Ｈ１の設定です。

290
面数（ＳＣ）や時計の針の進む時間の間隔（ＴＣ）の設定、スコアのクリアをします。

310
画面クリア

320
ＰＳＧの設定。

330
面数を＋１、時間の間隔を－0.5秒します。

340〜390
初期画面を描きます。

400〜410
レーダーの初期座標（ＸＰ、ＹＰ）の設定、時計のステップ数ＴＭのクリアをします。

420
レーダーの表示。

430〜440
割り込みの設定。

450
敵の初期値の設定。

470〜480
敵が向きをかえたとき、それからの移動方向（ＤＸ、ＤＹ）やステップ数（ＳＴ）を決めます。

490
もし、１面なら敵の座標に点を表示します。

500〜510
敵の移動。

520〜590
レーダーと敵の間の角度を求め、どの向きのスプライトを表示するかを決めます。

600
もし、１面なら490行で表示した点を消します。

620〜630
レーダーの移動。

640　レーダーの表示。

650

もし、敵が決められたステップ数を移動したなら、新たな方向を設定しなおすため470行へ、そうでなければ490行へとびます。

660

ＳＴＩＣＫ（０）と移動方向の関係のデータ。

670

ここから爆弾を発射したときの処理です。まず時計を止めます。

680〜760　爆発だ！

770　敵が出現！

780　当たったら810行へとびます。

790〜800

はずれた場合の処理です。1080行というのは太字でプリントするサブルーチンです。ここからゲームオーバーの処理ルーチンへとびます。

810〜850

スコアの加算、表示などです。

860

もし、10面がおわったなら900行へ。

870〜890

次の面にとびます。

900〜910

まいった！　ゲーム終了の処理ルーチンへとびます。

920

ここからはタイマ割り込みの処理です。時計の針が１回進むごとに呼び出されます。まず時計の針を１進めます。

930

もし、針が10回以上進んでいたなら時計を赤くし、音をかえます。

940

時計の音を鳴らします。

950

時計の表示。

960

まだ規定時間内ならゲームにもどります。

970

時間切れ！墜落音を出します。

980〜1050

ゲームオーバーの処理です。

1060

タイマーです。

1080〜1100

太字をプリントするルーチンです。

1110

ＰＳＧの初期データ。

```
100 SCREEN 2,2:COLOR 15,4,4
110 OPEN"grp:" FOR OUTPUT AS #1
120 FOR N=1 TO 8:READ XM(N),YM(N):NEXT N
130 FOR R=0 TO 13:READ X$:SOUND R,VAL("&h"+X$):N
EXT R
140 N=0
150 PI=ATN(1)*4:ST=2*PI/16
160 FOR XF=8 TO 248 STEP 16
170 CIRCLE(XF,8),7,15:CIRCLE(XF,8),1,15
180 LINE(XF,8)-(XF+COS(T)*6,8+SIN(T)*6),15
190 SP$=""
200 FOR L=N*16 TO N*16+7:SP$=SP$+CHR$(VPEEK(L)):
NEXT L
210 FOR L=N*16+256 TO N*16+256+7:SP$=SP$+CHR$(VP
EEK(L)):NEXT L
220 FOR L=N*16+8 TO N*16+15:SP$=SP$+CHR$(VPEEK(L
)):NEXT L
230 FOR L=N*16+256+8 TO N*16+256+15:SP$=SP$+CHR$
(VPEEK(L)):NEXT L
240 SPRITE$(N*4)=SP$
250 N=N+1:T=T+ST
260 NEXT XF
270 H1=2000
280 '
290 SC=0:TC=330:S1=0
300 '
310 CLS:COLOR 15,4,4
320 SOUND 7,&H3E
330 SC=SC+1:TC=TC-30
340 PSET(16,4),4:PRINT #1,USING"SCENE ##";SC
350 PSET(144,4),4:PRINT #1,USING"SCORE #####";S1
360 FOR N=1 TO 100
370  X=RND(1)*255:Y=RND(1)*191
380  PSET(X,Y),15:PSET(X+2,Y-1),15:PSET(X+4,Y),1
5:PSET(X+6,Y-1),15:PSET(X+8,Y),15
390 NEXT N
400 XP=120:YP=90
410 TM=0
420 PUT SPRITE 2,(100,0),3,12*4
430 ON STRIG GOSUB 670:STRIG(0) ON
440 ON INTERVAL=TC GOSUB 920:INTERVAL ON
450 XE=RND(-TIME)*256:YE=RND(-TIME)*192
460 '
470 ST=INT(RND(-TIME)*100)
480 DX=RND(1)*6-3:DY=RND(1)*6-3:CT=0
490  IF SC=1 THEN PSET((XE+8)MOD 256,(YE+8)MOD 1
92),4
```

```
500  XE=(XE+DX+256)MOD 256:YE=(YE+DY+192)MOD 192
510  CT=CT+1
520 XS=SGN(XP-XE):YS=SGN(YP-YE)
530  IF XP=XE OR YP=YE THEN 650
540  TH=ATN(ABS(YP-YE)/ABS(XP-XE))
550 ON (YS+1)*3+XS+1+1 GOTO 560,590,570,590,590,
590,580,590,590
560  TH=-PI+TH:GOTO 590
570  TH=-TH:GOTO 590
580  TH=PI-TH:GOTO 590
590  Q=(INT(TH/(2*PI)*16)+8)*4
600  IF SC=1 THEN PSET((XE+8)MOD 256,(YE+8)MOD 1
92),15
610 '
620  DR=STICK(0)
630  XP=(XP+256+XM(DR)*8)MOD 256:YP=(YP+192+YM(D
R)*8)MOD 192
640  PUT SPRITE 1,(XP,YP),15,Q MOD 64
650  IF CT=ST THEN 470 ELSE 490
660 DATA 0,-1 ,1,-1 ,1,0 ,1,1 ,0,1 ,-1,1 ,-1,0 ,
-1,-1
670 INTERVAL OFF
680 SOUND 7,&H37:SOUND 6,21:SOUND 13,0
690 FOR R=1 TO 20
700  CIRCLE(XP+8,YP+8),R,9
710 NEXT R
720 SOUND 6,2:SOUND 13,0
730 PUT SPRITE 0,(XE,YE),11,0
740 FOR R=1 TO 20
750  CIRCLE(XP+8,YP+8),R,4
760 NEXT R
770 PUT SPRITE 0,(255,192),0,0
780 IF (XP-XE)^2+(YP-YE)^2<400 THEN 810
790 X0=108:Y0=70:D1$="MISSED!":GOSUB 1080
800 GOTO 980
810 X0=80:Y0=70:D1$="ENEMY DESTROYED!":GOSUB 108
0
820 BN=SC*(16-TM)*10:S1=S1+BN
830 D1$="BONUS POINT ######*10=####"
840 PSET(40,85),4:PRINT #1,USING D1$;SC,16-TM,BN
850 PSET(41,85),4:PRINT #1,USING D1$;SC,16-TM,BN
860 IF SC=10 THEN 900
870 X0=80:Y0=100:D1$="LET'S PLAY NEXT.":GOSUB 10
80
880 GOSUB 1060
890 RETURN 310
900 X0=80:Y0=40:D1$="まいった! あんたは えらい!":GOSUB 1080
910 GOTO 1000
920 TM=TM+1
930 IF TM>10 THEN CL=9:SOUND 0,128:SOUND 1,0 ELS
E CL=3:SOUND 0,0:SOUND 1,1
940 SOUND 13,0
950 PUT SPRITE 2,(100,0),CL,((TM+12)MOD 16)*4
960 IF TM<16 THEN RETURN
970 SOUND 7,&H37:SOUND 13,0
980 X0=100:Y0=90:D1$="GAME OVER":GOSUB 1080
990 GOSUB 1060
1000 IF S1>H1 THEN H1=S1
1010 PSET (20,180),4:PRINT #1,USING"TODAY'S HIGH
 SCORE :######";H1
1020 X0=80:Y0=100:D1$="TRY AGAIN?(y/n)":GOSUB 10
80
1030 KY$=INPUT$(1)
1040 IF KY$="y" THEN IF TM<16 THEN RETURN 290 EL
SE 290
1050 IF KY$="n" THEN END ELSE 1030
1060 FOR TT=1 TO 1000:NEXT TT:RETURN
1070 '
1080 PSET(X0,Y0),4:PRINT #1,D1$;
1090 PSET(X0+1,Y0),4:PRINT #1,D1$;
1100 RETURN
1110 DATA 0,1,0,0,0,0,15,3e,10,0,0,0,80,0
```

ポイ大尉

●STORY

ここはノルマドロ共和国中央軍事科学研究所。現在次期主力戦闘機M 174 Sの開発に全力を注いでいる。そしてそれは初のテスト飛行を明日に控えた晩のことだった。その日は夕方から降り出した雨が晩には嵐に変わり、激しい雨と風が吹き荒れていた。そのなかを訓練飛行に飛びたった我が軍の戦闘機が、運悪くもこの研究所の敷地内に墜落してしまったのだ。それを他国の侵略と解釈した防衛システムのマザーコンピュータは第二防衛システムを作動させてしまった。ところがこの防衛システムは設置されてからすでに10年、部品の老朽化も目だってきたうえにこのシステムにとって実はこれがはじめての作動だった。研究所に閉じ込められた警備兵らはシステムの停止を幾度となく試みたが、すべての停止機能が働かなくなっていた。夜遅くになって起こったことだったので、修理のできる技術者はまったくいない。研究所のまわりには、地上は地雷群があり、低空には浮遊物がうごめいている。閉じこめられた警備兵らを助けるために、そしてM 174 Sを確保するためにわれわれはさまざまな方法で研究所にたどり着くことを試みたがことごとく失敗してしまい、残るは生身の人間によるパラシュート降下しか残されていない。そこで過去、空挺部隊で活躍し、現在は特殊工作隊指揮官ポイ大尉が選ばれた。

●遊び方

飛行機がポイ大尉を乗せて上空を飛びまわっています。スペースキーを押すことでその飛行機から大尉が降下し始めます。なおいつまでもためらっていると強制的に落とされます。降下し始めたら彼をカーソルキーの下左右でうまく操作し、浮遊物に触れないように気をつけて研究所のビルの前に無事たどり着かせてください。浮遊物に触れたり、研究所のビルの前からはずれたりすると、死亡となり残数が減ります。残数がなくなったり、スコアが9998点を超えたりしたとき、ＧＡＭＥ　ＯＶＥＲとなります。もっとも、ポイ大尉が何人もいるのも変な話ですが……。

●大尉と浮遊物の衝突について

浮遊物の移動処理ルーチンの初めと終わりの部分に大尉との衝突チェックがあります。これは大尉の移動、衝突チェック、浮遊物の移動、衝突チェックを繰り返し行うことで、取りこぼしなく衝突の確認ができます。衝突はＸ座標の差とＹ座標の差がある一定の範囲（ここではともに10ドット）内にあるとき、重なって表示される箇所ができる状態を衝突と考えて処理しているのであります。ここでたまにＦＯＲ－ＮＥＸＴ内にいてもかまわずその外に飛び出て帰ってこないプログラムを作る人がいますが、これはなるべく避けてください。このプログラムではＮという変数を作り、浮遊物と大尉の衝突が確認されたときにこの変数を０にすることで、衝突したという判断ができるようにさせております。

●主な変数リスト

H	ハイスコア
S	スコア
R	残数
G	ラウンド数
N	大尉生存フラグ （生＝1、死＝0）
Q	敵の最低移動ドット数
P	強制降下待機カウンター この値が0なら強制降下
X	降下中の大尉のＸ座標
Y	降下中の大尉のＹ座標 この値が＞0なら降下中 ＝0なら飛行機の中
U	研究所のあるＸ座標
Z	飛行機のＸ座標
A（i）	浮遊物の横の位置
C（i）	浮遊物の移動ドット数

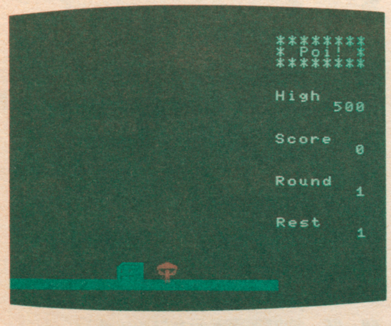
i＝0のとき　最上段の浮遊物
1のとき　二段目の浮遊物
2のとき　三段目の浮遊物
3のとき　最下段の浮遊物

●プログラムの構造

10～11	初期設定
12～13	キャラクタパターン設定
14～16	スプライトパターン設定
17～18	ゲームスタート音の登録
19	スタート時の変数初期設定 （リプレイ時の飛び先）
20	全スプライトの表示停止
21	スタート音実行
22	スタート音終了待ち
23～34	プレイ画面表示
35～37	強制降下待機カウンター、敵の最低移動ドット、生存フラグの設定、飛行機と研究所のビルの表示
38～39	浮遊物の表示
40～65	ゲーム　メイン　ルーチン
40	大尉降下中なら46へ
41～42	飛行機の移動
43	スペースキーが押されず、強制落下させる時間でもないなら、55へ
44～45	大尉出現処理、55へ
46	飛行機未表示なら49へ
47～48	飛行機の移動と表示、ただし画面範囲外なら消去
49～53	キー入力処理
54	着地なら77へ
55～64	浮遊物の移動処理
55	今、表示されているなら58へ
56	表示位置算出、61へ
57	この浮遊物のＹ座標算出
58	大尉と衝突したら62へ
59	移動先算出
60	表示範囲外のときの処理
61	大尉と衝突してないなら63へ
62	生存フラグを０へ
63	浮遊物表示
65	生存中なら40へ
66～71	死亡時の処理
72～74	9998点超えのメッセージ
75～76	ゲームオーバー処理、19へ
77～84	着地処理
77	研究所前から外れたなら66へ
78～82	ラウンドクリア処理
83	スコア処理
84	9998点超えなら72へ、それ以外はラウンドを更新し20へ
85～124	スプライトパターンデータ
85～88	降下中のポイ大尉
89～92	万歳ポーズ（両手上げ）
93～96	万歳ポーズ（両手下げ）
97～100	研究所ビル1
101～104	研究所ビル2
105～108	飛行機
109～112	最上段の浮遊物
113～116	二段目の浮遊物
117～120	三段目の浮遊物
121～124	最下段の浮遊物

```
10 SCREEN 1,2,0:COLOR15,1,1:CLS:KEY OFF:WIDTH32
11 DEFINT A-Z:DIM A(3),C(3):H=500
12 VPOKEBASE(6)+24,&HC1
13 FORI=0 TO7:VPOKEBASE(7)+1536+I,&HFF:NEXTI
14 FORI=0 TO9:A$="":FORJ=0 TO31
15 READB$:A$=A$+CHR$(VAL("&H"+B$)):NEXTJ
16 SPRITE$(I)=A$:NEXTI
17 A$="T180V9O4R4R8E4G8O5C4L4DCO4BFAGFE8D8D1"
18 B$="T180V9O4R4L8R8AR8AR8AAABABCBCCCBCCBCCC1"
19 G=1:R=2:S=0
20 FORI=0 TO9:PUT SPRITEI,(0,209):NEXTI
21 CLS:PLAYA$,B$:LOCATE12,9:PRINT"Round";G
22 IFPLAY(0) GOTO22 ELSECLS:I=RND(-TIME)
23 LOCATE0,23:PRINT"タタタタタタタタタタタタタタタタタタタタタタタタ";
24 LOCATE24,0:PRINT"********"
25 LOCATE24,1:PRINT"* Poi! *"
26 LOCATE24,2:PRINT"********"
27 LOCATE24,5:PRINT"High"
28 LOCATE26,6:PRINT USING"######";H
29 LOCATE24,9:PRINT"Score"
30 LOCATE26,10:PRINT USING"######";S
31 LOCATE24,13:PRINT"Round"
32 LOCATE26,14:PRINT USING"######";G
33 LOCATE24,17:PRINT"Rest"
34 LOCATE26,18:PRINT USING"######";R
35 P=80:Q=12+(G<4)*(8-G*2):Y=0:Z=176:N=1
36 PUT SPRITE5,(Z,0),15:U=INT(RND(1)*23)*8
37 PUT SPRITE3,(U,170),7:PUT SPRITE4,(U,170),5
38 FORI=0 TO3:A(I)=INT(RND(1)*24)*8
39 C(I)=(1-I*2+INT(I/2)*4)*INT(RND(1)*4+Q):NEXT
40 IFY>0 GOTO46
41 P=P-1:Z=Z-8-(Z=0)*184
42 PUT SPRITE5,(Z,0),15:PLAY"O3E64"
43 IFSTRIG(0)=0 ANDP>0 GOTO55
44 X=Z:Y=16:PUT SPRITE0,(X,Y),15
45 PLAY"O4A64F64":GOTO55
46 IFZ<0 GOTO49
47 Z=Z-8:IFZ<0 THENPUT SPRITE5,(0,209):GOTO49
48 PUT SPRITE5,(Z,0),15
49 ONSTICK(0)+1 GOTO53,53,51,51,51,50,52,52,52
50 Y=Y+2:GOTO53
51 X=X+8+(X=176)*8:GOTO53
52 X=X-8-(X=0)*8
53 Y=Y+3:IFY>169 GOTO77
54 PUT SPRITE0,(X,Y),15:PLAY"O2B64"
55 FORI=0 TO3:IFA(I)<177 GOTO57
56 A(I)=88-SGN(C(I))*88:B=I*24+72:GOTO61
57 B=I*24+72
58 IFABS(X-A(I))<11 ANDABS(Y-B)<11 GOTO62
59 A(I)=A(I)+C(I)
60 IFA(I)<0 ORA(I)>176 THENA(I)=255:B=209:GOTO63
61 IFABS(X-A(I))>10 ORABS(Y-B)>10 GOTO63
62 N=0
63 PUT SPRITE6+I,(A(I),B),8-I
64 NEXTI
65 IFN GOTO40
66 PUT SPRITE5,(0,208)
67 PLAY"O2V11C8V9C4V8C4V5C8","O2R8V10E4V7E4V6E8"
68 FORI=14 TO6 STEP-1:PUT SPRITE0,(X,Y),I
69 FORJ=1 TO99:NEXTJ,I
70 PUT SPRITE0,(0,209):R=R-1:IFR<0 GOTO73
71 FORI=1 TO2499:NEXTI:GOTO20
72 H=9999:LOCATE3,14:PRINT"How great you are!"
73 IFS<=H GOTO75
74 H=S:LOCATE3,11:PRINT"Your score is best."
75 LOCATE7,9:PRINT"GAME OVER"
76 FORI=1 TO3999:NEXTI:GOTO19
```

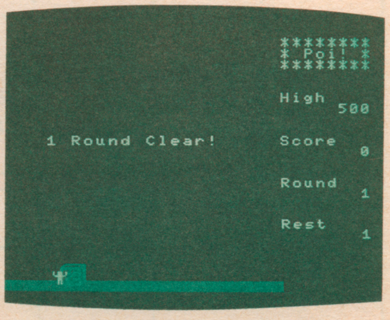
```
77  Y=170:IFX<U-8 ORX>U+8 GOTO66
78  PUT SPRITE5,(0,208):PUT SPRITE0,(0,209)
79  LOCATE4,9:PRINTG;"Round Clear!"
80  FORI=0 TO6:PLAY"O3F#16R32F4E64":FORJ=1 TO2
81  PUT SPRITEJ,(X,Y),15:FORL=0 TO199+J*100
82  NEXTL:PUT SPRITEJ,(0,209),15:NEXTJ,I
83  S=S+P+8-ABS(X-U)
84  IFS>9998 GOTO72 ELSEG=G+1:GOTO20
85  DATA03,0F,3F,7F,7F,40,23,33
86  DATA2B,1F,03,03,03,02,02,02
87  DATA80,E0,F8,FC,FC,04,88,98
88  DATAA8,F0,80,80,80,80,80,80
89  DATA00,00,00,00,00,00,13,13
90  DATA13,0F,03,03,03,02,02,02
91  DATA00,00,00,00,00,00,90,90
92  DATA90,E0,80,80,80,80,80,80
93  DATA00,00,00,00,00,00,03,03
94  DATA03,0F,13,13,13,02,02,02
95  DATA00,00,00,00,00,00,80,80
96  DATA80,E0,90,90,90,80,80,80
97  DATA1F,20,40,FF,80,80,9C,9C
98  DATA80,80,80,9C,9C,80,80,FF
99  DATAFF,03,05,F9,09,09,E9,E9
100 DATA09,09,09,E9,E9,0A,0C,F8
101 DATA00,00,0F,00,7F,42,42,42
102 DATA7F,7F,42,42,42,7F,7F,00
103 DATA00,00,FA,06,F6,16,16,16
104 DATAF6,F6,16,16,16,F4,F0,00
105 DATA00,07,08,48,4D,5F,5F,5F
106 DATAFF,58,5F,5F,5F,4F,0F,07
107 DATA0F,1F,BF,FF,FE,FC,FE,F1
108 DATAFF,0F,FF,FF,FE,FC,F8,F0
109 DATA07,1E,3F,7F,77,FB,FD,FD
110 DATAFD,FC,FC,7B,67,3F,1F,07
111 DATAFC,CE,77,7B,BB,BD,AB,AB
112 DATA75,F3,F7,F7,F7,EF,9E,FC
113 DATA00,00,07,1C,3C,7F,7F,FF
114 DATA7F,7F,3F,1F,07,00,00,00
115 DATA00,00,FF,33,33,FF,FF,FF
116 DATAFF,FF,FF,FF,FF,00,00,00
117 DATA07,1E,3F,7F,77,FB,FD,FD
118 DATAFD,FC,FC,7B,67,3F,1F,07
119 DATAFC,CE,77,7B,BB,BD,AB,AB
120 DATA75,F3,F7,F7,F7,EF,9E,FC
121 DATA00,00,07,1C,3C,7F,7F,FF
122 DATA7F,7F,3F,1F,07,00,00,00
123 DATA00,00,FF,33,33,FF,FF,FF
124 DATAFF,FF,FF,FF,FF,00,00,00
```



⬆降下失敗！研究所に届かなかった…



⬆下方向に加速するのは一体どういうしくみ？



⬆研究所に無事着陸しました。ただ今から……

迷路ゲーム

●遊び方

ここは行く手の見えない迷路。この暗闇のつづく迷路で、あなたはひたすら〝勘〟だけを頼りにゴール目指して突き進んでください。カーソルキーを押して移動することで、その周囲の様子が見えてきます。そうして、通路に沿って一歩進んでみると、二手に通路が分かれていたり、行き止まりになっていたりすることもあります。四苦八苦しながら、Ｔｉｍｅがなくなるまでの60秒間のうちに何とかゴールにたどり着いてください。それでは健闘を祈ります。

●迷路の作成方法について

まず、方眼用紙を思い浮かべてください。そして、その方眼用紙の太い線を頭の中でしっかり引いておきます。それでは、本題である迷路の作成方法について説明いたします。先程の方眼用紙上にあるひとつのマスを始点とします。そこを中心に隣接する四方のマスを見渡し、まだ通路としてチェックされていないマス、つまり四方の黒線がすべて存在するマスを見つけます。そして、見つけ出したマスを通路にある黒線を消去して、このマスを通路としてつなぎます。そして、今度はこのマスを中心に同様のことを繰り返します。なお、隣接する四方のマスがすべて通路としてチェックされていた（つまり、行き止まりとなってしまった）場合は、以前に通路としてチェックされているマスを逆にたどり、また別方向に通路を拡張可能なマス（つまりまだ四方がすべて通路としてチェックされていない）を探して、そこからまた新しい通路を作っていきます。そして、どこまで通路を戻っても新たに通路が作れなくなったとき、迷路作成は終了したことになります（だってもう空白部分は残っていないわけですからね）。

これが、一本の通路から枝葉のごとく通路をひろげ、迷路として完成させるテクニックと言えます。それでは、この方眼用紙をいかにしてこのプログラムが扱うかというと、まずマスも、その周囲の線も、配列の一構成要素と考えます。つまり、黒線を壁とすることで壁も通路もマスとしてひとつの同じブロックとして取り扱います。Ｍ（ｘ、ｙ）という配列変数がそれです。この配列変数Ｍ（ｘ、ｙ）のｘとｙがともに偶数（但し、１＜ｘ＜25、１＜ｙ＜23）であるところが方眼用紙上のマスで、その他が黒線です。以上が迷路作成の基本的な考え方です。

しかし、上の原理どおりにプログラムを組むと遅くて使い物になりません。（通路を戻る、という作業が非常に時間を食います）。そこでこのプログラム上では、多少手抜きをしています（わかりますか？）が、これもすべて時間短縮によるものですので、勘弁してください。

●改造点について

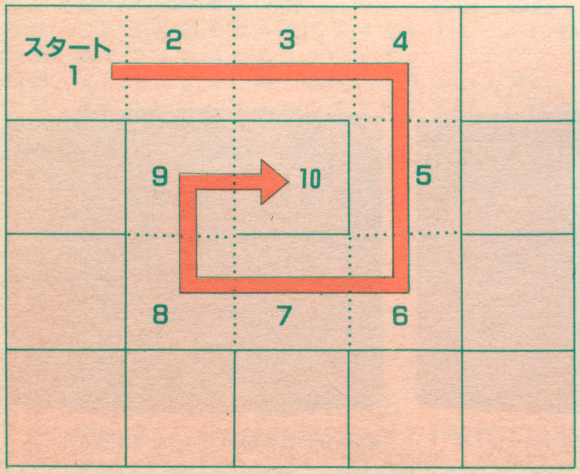
ジョイスティックを使用したい方は、17行の〝ＳＴＲＩＧ（0）〟を〝ＳＴＲＩＧ（1）〟にし、21行の〝ＳＴＩＣＫ（0）〟を〝ＳＴＩＣＫ（１）〟にすることで、ジョイスティック１を使うことができます。また、19行にある二つの〝60〟を適当な同じ数値に変えることで、制限時間がその変えた数値分の秒数になります。

●主な変数リスト

H	ハイスコア
S	スコア
X	キャラクタの横の位置（１＜Ｘ＜25）
Y	キャラクタの縦の位置（１＜Ｙ＜23）
V	横の移動方向（左＝－１、右＝１、現在地＝０）
P	残りの時間（秒）
W	縦の移動方向（左＝－１、右＝１、現在地＝０）
Ｍ（ｘ、ｙ）	迷路の構造（通路＝０、壁＝１）

●プログラムの構造

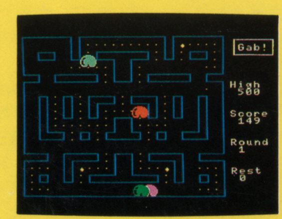
10—11	初期設定
12	キャラクタパターン設定
13	迷路作成、初期画面表示
14—16	キャラクタの位置設定、スタート待ち画面の作成
17	スペースキー入力待機
18	タイマーの初期化
19—26	ゲーム　メイン　ルーチン
19	残り時間算出、０なら27へ
20	残り時間の表示
21	キー入力、入力無しなら19へ
22	移動方向算出
23	移動先が壁なら音を出し19へ
24	移動元のキャラクタを消す
25	キャラクタの移動、音を出す、周囲の表示
26	ゴールに到着なら29、他19へ
27—28	タイムアウト処理、ハイスコア処理、スコア初期化、32へ
29—31	ゴールイン処理、スコア処理
32—34	迷路のオープン表示
35	迷路作成、初期画面表示、14へ
36—38	周囲の表示サブルーチン
39—57	迷路作成サブルーチン
58—66	初期画面表示サブルーチン



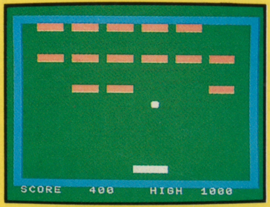
たとえばこのようになってしまった場合、迷路を逆にたどって（この例では10→９→８…）他に通路を作れるところを探し、そこで新たな通路を作る。このようにして、もう他にどこにも通路が作れなくなるまで続ける。

```
10 SCREEN1,2,0:COLOR11,1,1:KEY OFF:DEFINTA-Z
11 DIMM(26,24):WIDTH32:VPOKEBASE(6)+24,81
12 FORI=0 TO7:VPOKEBASE(7)+1536+I,&HFF:NEXTI
13 H=100:S=0:GOSUB58:GOSUB39
14 X=2:Y=2:GOSUB36:LOCATE2,2:PRINT"●"
15 LOCATE25,22:PRINT"Goal"
16 LOCATE5,0:PRINT"Strike Space key.":LOCATE5,0
17 IFSTRIG(0) THENPRINTSTRING$(17," ") ELSE17
18 LOCATE27,16:PRINT"TIME":TIME=0
19 P=60-INT(TIME/60):IFP=<0 GOTO27
20 LOCATE29,17:PRINTUSING"##";P
21 I=STICK(0):IFI=0 GOTO19
22 V=(I=7)-(I=3):W=(I=1)-(I=5)
23 IFM(X+V,Y+W) THENPLAY"O4F64":GOTO19
24 LOCATEX,Y:PRINT" ":X=X+V:Y=Y+W
25 LOCATEX,Y:PRINT"●":PLAY"O4B64":GOSUB36
26 IFX=24 ANDY=22 GOTO29 ELSE19
27 LOCATE29,17:PRINT"OUT":IFS>H THENH=S
28 S=0:PLAY"O4L16BR32AR32GR64FR64EDC":GOTO32
29 PLAY"T200O3R2L32AR8":LOCATE27,19:PRINT"Nice!"
30 PLAY"AR8AR4GR8GR8GR4AR8AR8AR8AR8AR8AR8AR3A1"
31 S=S+P:IFS>9999 THENS=S-9999
32 FORJ=2 TO22:FORI=2 TO24
33 IFM(I,J)=0 THENLOCATEI,J:PRINT" "
34 NEXTI,J:LOCATEX,Y:PRINT"●"
35 GOSUB39:GOSUB58:GOTO14
36 FORL=0 TO3:I=(L=0)-(L=2):J=(L=1)-(L=3)
37 IFM(X+I,Y+J)=0 THENLOCATEX+I,Y+J:PRINT" "
38 NEXTL:RETURN
39 I=RND(-TIME):LOCATE7,0:PRINT"Wait a minute."
40 FORI=1 TO25:FORJ=1 TO23:M(I,J)=1:NEXTJ,I
41 FORI=0 TO26:M(I,0)=0:M(I,24)=0:NEXTI
42 FORJ=1 TO23:M(0,J)=0:M(26,J)=0:NEXTJ
43 M(2,2)=0:FORI=2 TO24 STEP2:FORJ=2 TO22 STEP2
44 X=I:Y=J:IFM(I,J) GOTO54
45 L=INT(RND(1)*4):N=L
46 ONL GOTO48,49,50,47,48,49
47 V=0:W=1:L=1:GOTO51
48 V=-1:W=0:L=2:GOTO51
49 V=0:W=-1:L=3:GOTO51
50 V=1:W=0:L=0
51 IFL=N GOTO54
52 IFM(X+V*2,Y+W*2)=0 GOTO46
53 X=X+V*2:Y=Y+W*2:M(X-V,Y-W)=0:M(X,Y)=0:GOTO45
54 NEXTJ,I
55 FORI=2 TO24 STEP2:W=(I>2)*2:FORJ=2 TO22 STEP2
56 IFM(I,J) THENM(I,J)=0:M(I+W+1,J)=0
57 NEXTJ,I:PLAY"T255O4A8F8":RETURN
58 CLS:LOCATE26,1:PRINT"******"
59 LOCATE26,2:PRINT"*MAZE*"
60 LOCATE26,3:PRINT"******"
61 LOCATE27,7:PRINT"High"
62 LOCATE28,8:PRINTUSING"####";H
63 LOCATE27,11:PRINT"Score"
64 LOCATE28,13:PRINTUSING"####";S
65 FORJ=1 TO23:LOCATE1,J:FORI=1 TO25
66 PRINT"タ";:NEXTI,J:RETURN
67 '
68 ' program:maze, poi & hipman
69 '     are all produced by
70 ' T. Someya              (C) 1987
71 '
```

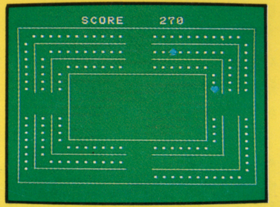
★編集長：田口旬一　★デザイン：スタジオＢ４　★プログラム：飯沼健＋染谷哲人



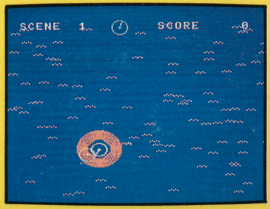
ヒップマン



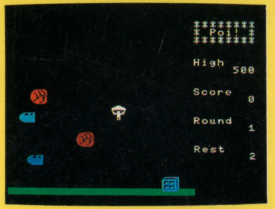
ブロックくずし



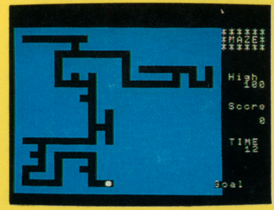
ヘッドオン



BOMB//



ポイ大尉



迷路ゲーム

エムエスエックスマガジン第５巻第11号通巻48号昭和62年11月１日発行（毎月１回１日発行）昭和59年２月６日第３種郵便物認可